

Adaptive Mutation Strategies for Evolutionary Algorithms

Stephan Blum^{1*}, Romanas Puisa², Jörg Riedel¹, Marc Wintermantel³

¹ DYNARDO – Dynamic Software and Engineering GmbH, Weimar, Germany

² Faculty of Mechanics, Vilnius Gediminas Technical University, Vilnius, Lithuania

³ EVEN - Evolutionary Engineering AG, Zurich, Switzerland

Abstract

The purpose of this study is to investigate and elaborate various strategies of adapting mutation within evolutionary algorithms. Since the early development of evolutionary optimization, adaption has always been a topic of research and numerous strategies with different levels and types of adaptation have been proposed.

Our motivation is the application of evolutionary algorithms for solving real-world optimization problems, which often requires a problem-specific algorithm design. The complex customization of the algorithms to the specific problem either needs a specialist or limits the field of application to mainly scientific research. But there is a growing demand for up-to-date optimization software, applicable by a non-specialist within an industrial development environment. A possible way to supply this demand are intelligent algorithms containing a mechanism to modify the parameters without external control. This process of dynamic adaptation happens while the algorithm is searching for the problem solution.

Within this study we investigate different adaptation methods regarding their effect on optimization results. The discussion of these results shall give an indication towards promising self-adapting strategies for practical application.

Keywords: Evolutionary Algorithm, Adaptive Mutation, Fitness Landscape Analysis, Probability Density Modulation

*Kontakt: Dipl.-Ing. Stephan Blum, DYNARDO – Dynamic Software and Engineering GmbH, Luthergasse 1d, D-99423 Weimar, E-Mail: stephan.blum@dynardo.de

1 Introduction

Adaption within evolutionary algorithms (EAs) reflects the attempt to mimic processes of natural evolution. State of the art EA implementations often require a comprehensive algorithmic knowledge from the user in order to choose appropriate strategy parameters for solving a specific optimization problem. Even for an expert, the parameter configuration for an optimal performance is hard to find. The idea of adaption is to change these parameters regarding the current state of search. A survey on adaption in evolutionary computation is given in (Hinterding u. a. (1997)).

In this paper two methods of adapting mutation parameters based on different criteria are presented. The first strategy analyzes the *fitness frequency distribution* in order to determine the current search state. A second method evaluates successful mutations and modulates probability density functions of mutation steps. The proposed methods are applied to different test problems and evaluated regarding their performance compared to the use of static mutation parameters.

2 Fitness Landscape Analysis

The introduction of the *fitness landscape* concept (Wright (1992)) was intended to explain the dynamics of biological evolution. From the mathematical point of view, the fitness landscape (FL) represents a $(N + 1)$ -dimensional hypersurface, where N is the number of genes in a chromosome while the extra dimension stands for a fitness function F . The genes represent the components of a N -dimensional point P , and the fitness function is a map $F : P \rightarrow \mathbb{R}$. The objective of EAs is to search for the lowest or highest peak of FL while minimising or maximising, respectively. The shape of FL reflects a physical problem that can be inhomogeneous having several different homogeneous sub-regions. Within the scope of this paper we distinguish unimodal, multimodal, or flat sub-hypersurfaces.

At each FL region the search by EAs undergoes a corresponding state. The list of FL regions with appropriate EA search states is as follows:

1. **Flat or neutral** - Steady state with weak population replacement.
2. **Multimodal** - Varying with increasing and decreasing convergence.
3. **Unimodal** - Population brackets an optima; relatively steady in terms of the improvement/convergence.

The first EA search state (further on just *search state*) is most undesirable due to the weak replacement, since offspring populations have no improvement. Therefore the algorithm must jump out from this state as soon as possible. The second FL state is the case of multimodality. The search is inefficient due to the varying convergence. As a result, the overall search improvement can be quite small or even not exist at all (e.g. all the local optima have the same fitness values), therefore this FL state is also undesirable. At the third EA search state, the search converges to an optimum, so that it is a most desirable state of the search.

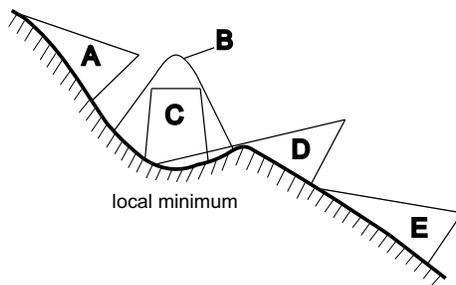


Abbildung 1: Creeping on FL (from the left to the right), 1D model. The lettered shapes denote the fitness distribution while minimising.

To jump out from undesirable states, some search strategies changing the EA behaviour have to be employed (Beyer und Deb (2001)). The adjustment of the search in a way of increasing search efficiency is the main purpose of the fitness landscape analysis. The actual search state reflects a FL region being explored. Statistical analysis methods based on the FL mean and variance are generally proposed for the analysis. Such methods allow to adjust the search by choosing the appropriate values for EA parameters. For example, those parameters can include: the crossover and mutation rates and types of crossover or mutation used. Therefore, the final purpose of the FL analysis is to draw the explicit relationship Σ between a search state S and EA parameters \mathbf{V} , as $\Sigma : S \rightarrow \mathbf{V}$. In the following section, we propose one point of view on the fitness landscape analysis and consequently establish Σ .

2.1 Fitness Frequency Distribution

Generally, a population represents a list of $(N + 1)$ -dimensional points. However, let's take advantage only of the $(N + 1)$ 'th dimension, i.e. the fitness value. So, we can also consider the population as a list of one dimensional points, where point values correspond to fitness values. For example consider the 1D fitness landscape and population fitness shapes creeping on it. Fig. 1 shows a quite idealised model of the population creeping on the FL. Analysing it from the left to the right, the population moving towards the local minimum (shape **A**) has quite a large diversity due to the high positive (the majority of offspring are better than their parents) replacement rate, i.e. the population converges to a local optimum. After having reached it (shape **B**), the population should be distributed symmetrical around it. Then, after several generations without improvement, the shape becomes flatter, i.e. more points with similar fitness values appear (shape **C**). Fortunately, some individuals get produced by genetic operators out of the local optimum, and the search starts moving down towards the better solutions (shape **D**). After several generations the population gains the big positive replacement, i.e. the search is efficient and converges (shape **E**).

The fitness shape of a population changes during the search and reflects the search or convergence state. To find the fitness shape, the *fitness frequency distribution* (FFD) histogram has to be constructed and appropriate statistic methods have to be used to investigate it. The FFD histogram is constructed from the fitness values t_i which are

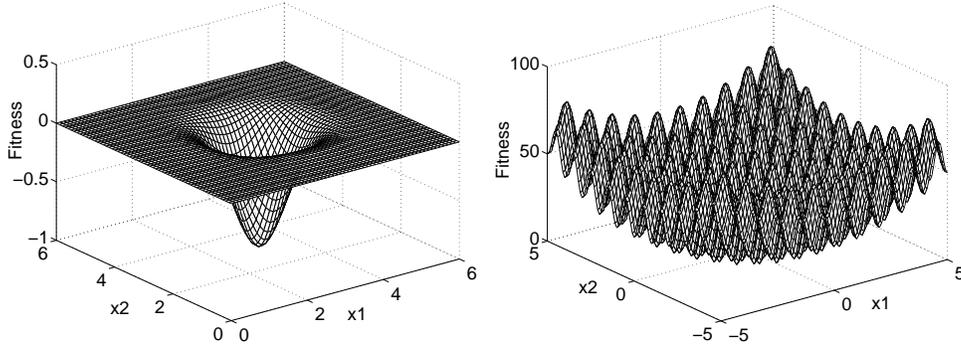


Abbildung 2: Easom's (left) and Rastrigin's test function.

divided into *class intervals* (Kenney und Keeping (1962)), where the frequency f_i of a certain fitness value t_i shows what part of a population it occupies, that is

$$f_i = \frac{n_i}{D}, \quad (1)$$

where D is the population size and n_i is the number of individuals in the population falling into the class interval with i th fitness value as a *class mark*.

If we take the frequency value of a best fitness value (BFF), we can also see the convergence efficiency of EAs since the BFF also reflects the FFD shape. To calculate the BFF the class interval with the best fitness value as a class mark is used. It follows the expression

$$BFF = \frac{1}{D} \sum_{i=1, i \neq b}^N \delta_i, \quad BFF \in [0, 1), \quad (2)$$

where

$$\delta_i = \begin{cases} 1, & t_i \in [t_b - \epsilon, t_b + \epsilon], \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

t_b is the best fitness value, and t_i is the fitness value of i th point. The bandwidth ϵ of the class interval is calculated as a percentage $r \in (0, 1]$ from the t_b magnitude

$$\epsilon = t_b \cdot r. \quad (4)$$

In this work we used $r = 0.05$ as 5% of the t_b value. In the following the BFF measures for Easom's and Rastrigin's (Fig. 2) test functions are presented. Fig. 3 shows how BFF values reflect the search state. Since $BFF \in [0, 1)$, it rises up to approx. 1 when search creeps on the flat and goes down to 0 when the search is most efficient. In case of the FL multimodality and EA convergence, BFF has small values with some peaks indicating temporal lingering at a local minimum (Fig. 4).

Summarising, BFF reflects the current search state and can be used to indicate its efficiency. The coming section discusses how BFF can be used for the mutation rate and mutation step adaptation.

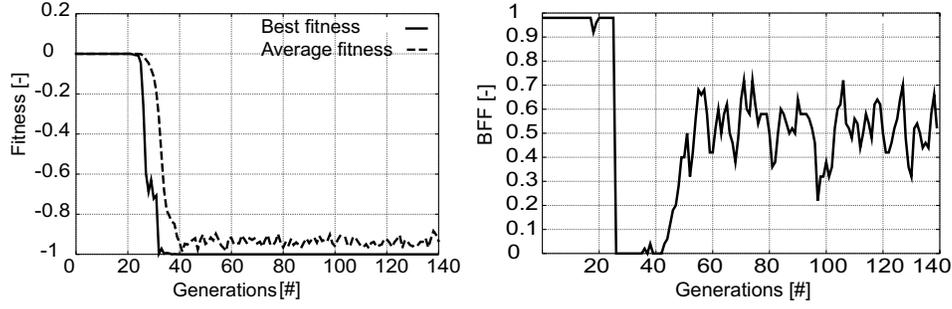


Abbildung 3: Plain-GA convergence behavior and corresponding BFF behavior for Easom's test-function.

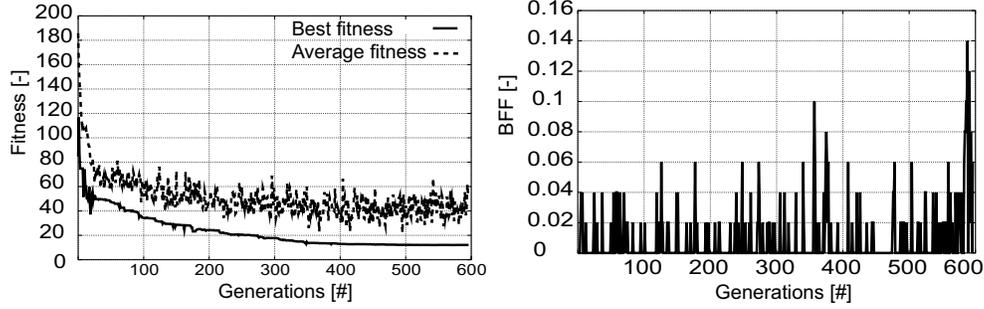


Abbildung 4: Plain-GA convergence behavior and corresponding BFF behavior for Rastrigin's test-function.

3 Mutation Adaption in Evolutionary Algorithms

3.1 Evolutionary Algorithm

In this section we outline the algorithm used in our work. We derive our evolutionary algorithm from the GAs (Holland (1975), Goldberg (1989), Bäck (1996)). The algorithm follows the common scheme of GAs however, instead of the classical binary genotype, where every gene g_i in a chromosome is defined as $\forall i : g_i \in \{0; 1\}$, we use the floating-point representation, $\forall i : g_i \in \mathbb{R}$ (Michalewicz (1992)).

The floating-point representation implies the use of appropriate genetic operators, since, for example, the classical binary mutation (Goldberg (1989)) is absolutely meaningless. The algorithm uses two crossover operators: *simple crossover* and *arithmetical crossover*. The simple crossover operator is a classical n -point crossover, where $n \geq 1$. When $n = 1$ we have a simple one-point crossover operator. The arithmetical crossover mates two parent individuals, \mathbf{g}_v^t and \mathbf{g}_w^t , and bears two children, \mathbf{g}_v^{t+1} and \mathbf{g}_w^{t+1} , in the following way

$$\mathbf{g}_v^{t+1} = \lambda \mathbf{g}_v^t + (1 - \lambda) \mathbf{g}_w^t, \quad (5)$$

$$\mathbf{g}_w^{t+1} = \lambda \mathbf{g}_w^t + (1 - \lambda) \mathbf{g}_v^t, \quad (6)$$

where $\lambda \in \mathbb{R}$ is a constant, which is randomly generated within the interval $\lambda \in [0, 1]$.

We use the *Gaussian mutation* operator introduced into *Evolution Strategies* (Rechen-

berg (1973)). Here every gene g_i is mutated with the mutation probability p_m as follows

$$g_i^{t+1} = g_i^t + N(0, \sigma), \quad (7)$$

where $N(0, \sigma)$ is the normal distribution with zero mean and the standard deviation σ . The mutation probability p_m is also called mutation rate, which stays constant in the classical GAs. However, the algorithm becomes more efficient when p_m varies depending on the algorithmic convergence. The following subsection discusses the p_m adaptation issue.

3.2 Mutation Rate Adaptation

The previous section has shown that BFF values indicate the efficiency of the search and can be used for changing its behaviour. Actually, the EA behaviour is changed by increasing or decreasing *population movement intensity* (PMI) depending on the search state. The mutation rate has a significant effect on PMI, but how can the mutation rate be changed?

In fact, the mutation rate should be increased in case of the flatness of FL or its big multimodality as defined in the basic search strategies (see section 2). The increase of the mutation rate would rise the PMI and increase the exploration of the fitness landscape (Spears (2000)). We chose a deterministic strategy where the mutation rate is proportional to the search efficiency criteria BFF. The mutation rate is calculated as

$$p_m^D = p_{avg} + \mu, \quad (8)$$

with the average mutation rate

$$p_{avg} = \frac{p_{max} + p_{min}}{2} \quad (9)$$

and

$$\mu = 3\sigma \left(\frac{\bar{f}^L}{\xi} \right), \quad \mu \in [0, 3\sigma], \quad (10)$$

where \bar{f}^L is an average BFF value within L generations, ξ is a *linking coefficient* and σ is the standard deviation of the Gaussian mutation-operator. Values for $\xi \in [1, 2]$ showed to produce reasonable results.

3.3 Mutation Step Adaptation

In this section, we present the scheme for the mutation step adaptation (MUSA). The formula (7) expresses the Gaussian mutation in a general form. Another general form is as follows

$$g_i^{t+1} = g_i^t + \sigma N(0, 1), \quad (11)$$

or

$$g_i^{t+1} = g_i^t + S_m N(0, 1), \quad (12)$$

where $S_m = \sigma$ is the mutation step for Gaussian mutation. Several MUSA approaches have been proposed (Hansen und Ostermeier (1996)), (Hansen und Ostermeier (2001)),

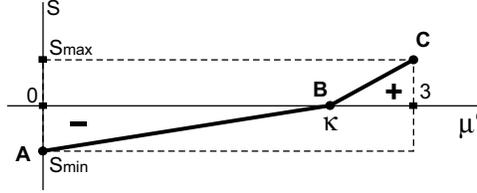


Abbildung 5: Mutation step adaptation scheme. The line segment **AB** decreases the mutation step value, meanwhile the segment **BC** increases it.

(Müller (2002)) and a comprehensive review of mutation step adaptation can be found in (Hinterding u. a. (1997)).

In the following we propose one more model for the S_m adaptation. The search strategies mentioned in section 2 state that the PMI must be increased in case of inefficient search. PMI is also directly proportional to S_m , so that by increasing S_m the search faster escapes infeasible regions of the fitness landscape. Seeking to minimise the number of tuning parameters for the used EA, the S_m is expressed through the already defined mean for normal distribution μ (10). However, the expression for μ calculation is transformed to μ' as discussed below. Fig. 5 shows the graphical interpretation of S_m adaptation. The relationship between μ' and S_m is expressed by two line segments AB and BC. If the mean $\mu' < \kappa$, then the mutation step is decreased until its minimum value S_{min} , otherwise S_m rises up. Mathematically it is expressed in the following way

$$S_m = \begin{cases} A_0 + A_1\mu', & \mu' \leq \kappa, \\ B_0 + B_1\mu', & \text{otherwise,} \end{cases} \quad (13)$$

where A_i and B_i are coefficients for line segments interpolation calculated as

$$A_0 = S_{min}, \quad A_1 = \frac{S_{avg} - S_{min}}{\kappa}, \quad (14)$$

$$B_0 = \frac{\kappa S_{max} - 3S_{avg}}{\kappa - 3}, \quad B_1 = \frac{S_{avg} - S_{max}}{\kappa - 3}, \quad (15)$$

where $S_{avg} = \frac{S_{min} + S_{max}}{2}$ is an average mutation step between the predefined minimum S_{min} and maximum S_{max} values. The parameter $\kappa \in (0, 3)$ is a threshold value, once it is reached by μ' the S_m is increased above its average value. We can calculate the increase probability p_s^+ for the mutation step S_m if the parameter κ is given. Explicitly this probability is expressed as

$$p_s^+ = 1 - \frac{\kappa}{3}, \quad (16)$$

consequently, the decrease probability p_s^- for the mutation step S_m is

$$p_s^- = 1 - p_s^+ \quad (17)$$

The μ' is derived from the formula (10) expressing a mean value for the probabilistic mutation rate adaptation:

$$\mu' = 3 \left(\frac{\bar{f}^L}{\xi} \right), \quad \mu' \in [0, 3] \quad (18)$$

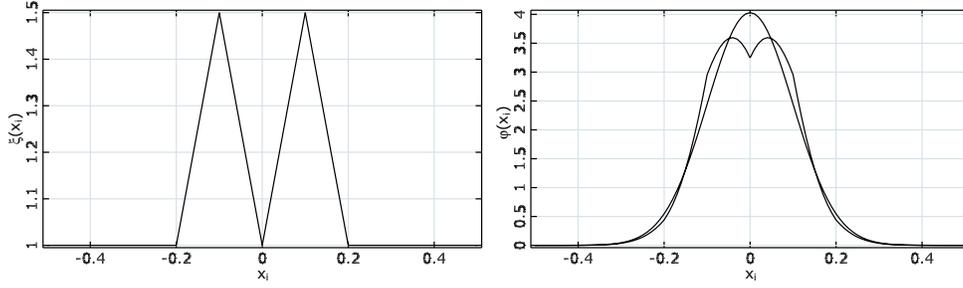


Abbildung 6: Modulation function $\xi(x_i)$ with $\delta_i = 0.1$ and $\sigma_i = 0.2$ (left) and probability density functions $\varphi^t(x_i)$ and $\varphi^{t+1}(x_i)$ (right).

This way of adaptation for the mutation step is based on the assumptions defined in section 2. That is, the mutation step increases in case of a flat FL, and rises down during the multimodality and quick convergence. The threshold κ also represents a μ' value before which the search is supposed to be either creeping on the multimodal FL or quickly converging.

3.4 Modulated Probability Density Mutation

The methods of adapting mutation parameters described so far are based on the analysis of the fitness landscape and are applied to Genetic Algorithms. The following method analyzes successful mutations and adaptively changes the distribution hypothesis for mutation steps.

The mutation of the gene g_i is defined as

$$g_i^{t+1} = g_i^t + X_i ; \quad i \in (1, n) \quad (19)$$

where X_i is a random variable with zero mean and standard deviation σ_i^t . At the initial stage a normal distribution is assumed for each random variable X_i . The corresponding *probability density function* (PDF) is given by

$$\varphi^t(x_i) = \frac{1}{\sigma_i^t \sqrt{2\pi}} \exp \left[-\frac{x_i^2}{2(\sigma_i^t)^2} \right] ; \quad t = 0 \quad (20)$$

The adaption takes place on component level and is based on the evaluation of successful mutations. For a successful mutated offspring individual, whose fitness is better than the fitness of its parent, the mutated genes and the realised mutation steps δ_i are identified. The probability density function of the random variable X_i corresponding to each mutated gene g_i , which has led to an improved fitness, is modulated to

$$\varphi_{mod}^t(x_i) = \varphi^t(x_i) (1 + \xi(x_i)) \quad (21)$$

where $\xi(x_i)$ is a symmetric modulation function which takes a value h if $|x_i| = \delta_i$ and becomes zero for $|x_i| > |\delta_i| \pm \sigma_i^t$. The modulated probability density function has to be

normalised according to

$$\varphi^{t+1}(x_i) = \varphi_{mod}^t(x_i) / \int_{-\infty}^{+\infty} \varphi_{mod}^t(x_i) dx_i \quad (22)$$

The resulting PDF $\varphi^{t+1}(x_i)$ still has zero mean due to the symmetric modulation. The new standard deviation of each random variable X_i can be calculated as

$$(\sigma_i^{t+1})^2 = \int_{-\infty}^{+\infty} x_i^2 \varphi^{t+1}(x_i) dx_i \quad (23)$$

Figure (6) illustrates both the modulation function $\xi(x_i)$ and the modulated probability density function. The method provides a self-adaptive strategy for changing the probability density of the mutation steps. The probability of mutation steps which led to an improvement is increased by the modulation function.

4 Simulation and Comparison

In this section we present the simulation for the mutation rate and mutation step adaptation. We also give the comparison between a Plain-EA and an EA search with the integrated adaptive approaches (Adaptive-EA).

4.1 Simulations of BFF-based Adaptation

The first test case (TC1) is Easom's test function presented in the previous section (see Fig. 2). The function is unimodal, where the descent to the global minimum covers a small area relative to the search space. Due to the small area, the probability to randomly generate a point inside this area is quite small. Therefore, a population often creeps on the flat FL at the beginning of the optimisation. This creeping must cause an increment in the mutation rate as has been defined. If the search finds that small area, a quick convergence towards the global minimum occurs, and the mutation rate can be decreased. During the stay at the global minimum, the search has no improvement, however due to applied mutation, some individuals get mutated outside of the optimum and the average fitness value is always changing. These fluctuations must cause the appropriate variance of the mutation rate because of the changing BFF. The results of the simulation are presented in Fig. 7. A mutation rate of $p_{max} = 0.5; p_{min} = 0.1$, a crossover rate $p_x = 0.5$ and a population size $D = 50$ was used.

The test function for the second test case (TC2) is Rastrigin's function (see Fig. 2). The simulation results for TC2 are illustrated in Fig. 8. A mutation rate of $p_{max} = 0.7; p_{min} = 0.1$, a crossover rate $p_x = 0.5$, a mutation step $S_{max} = 0.5; S_{min} = 0.1$ and a population size $D = 50$ was used. The results show a fast convergence from the beginning combined with a small BFF value. With proceeding convergence to the global minimum the BFF values are increasing. Compared with the results of the Plain-GA (see Fig. 4) the adaptive strategy produced improved results for this test function.

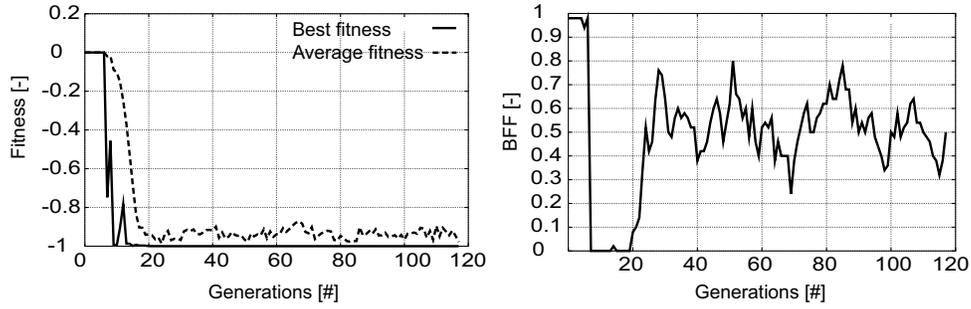


Abbildung 7: TC1: The Best and Average fitness (left) and BFF values while optimising Easom’s test function.

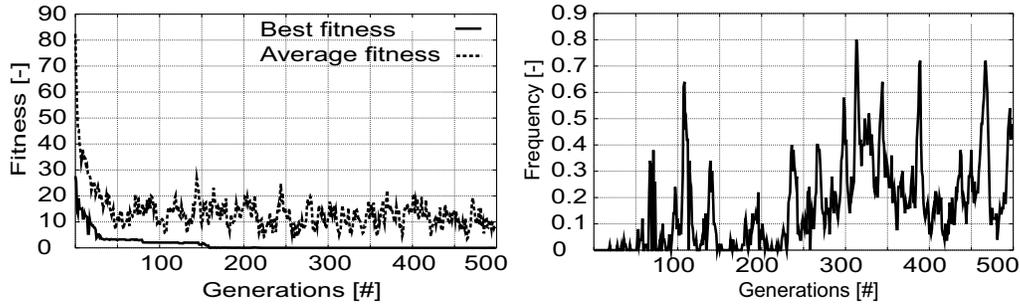


Abbildung 8: TC2: Best and Average fitness (left) and BFF values while optimising Rastrigin’s test function.

4.2 Simulations of Modulated PDF Adaptation

The adaptive mutation strategy of modulating probability densities for mutation steps is applied to both test problems as used in the previous section. In contrast to the BFF-based adaption, an evolution strategy without crossover is used. The replacement scheme conforms with the $(\mu + \lambda)$ strategy. The values $\mu = 1$, $\lambda = 30$ and mutation rate = 0.2 are used. The initial standard deviation is set to $\sigma_i^1 = 0.2$ for all n variables.

Figure 9 shows the average fitness of 10 different simulations optimizing Easom’s function (TC1). The logarithmic diagram reveals the the faster convergence and better approximation of the global optimum when applying the adaptive mutation strategy. The course of standard deviations σ_i^t is illustrated in Fig. 10 with the corresponding modulated probability density functions of parameter 5 for different generations. Because the global optimum is closely approximated after 20 generations, only small mutation steps lead to an improvement afterwards, hence the adaption mechanism emphasizes the probability of small mutation steps. The results for TC2 shows a similar performance of the adaption mechanism (see Fig. 11 and 12).

5 Conclusions

Both strategies presented in this paper have proven to be suitable for adaptively changing mutation paramters within Evolutionary Algorithms. The adaptive mutation based on

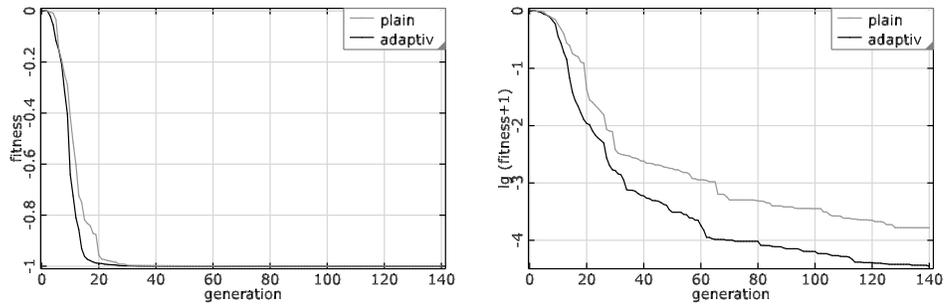


Abbildung 9: TC1: Fitness values of plain-EA and modulated pdf-mutation while optimising Easom's test function.

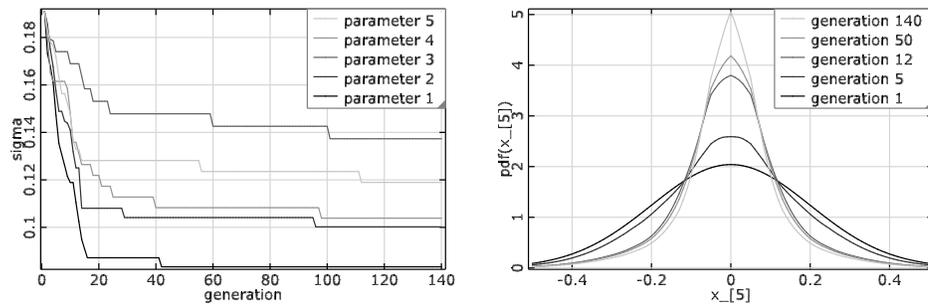


Abbildung 10: TC1: Course of standard deviations of mutation steps (left) and modulated pdf (right) while optimising Easom's test function.

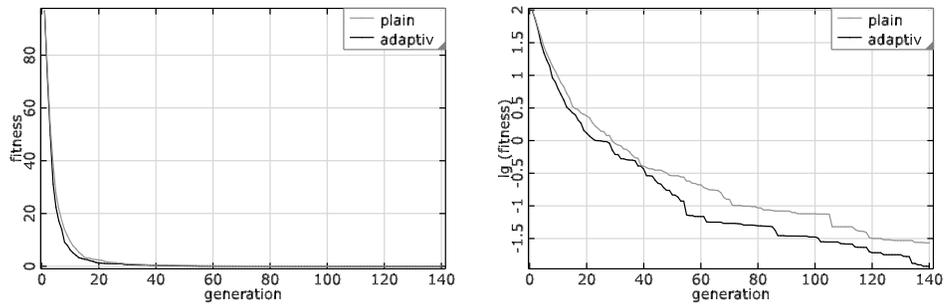


Abbildung 11: TC2: Fitness values of plain-EA and modulated pdf-mutation while optimising Rastrigin’s test function.

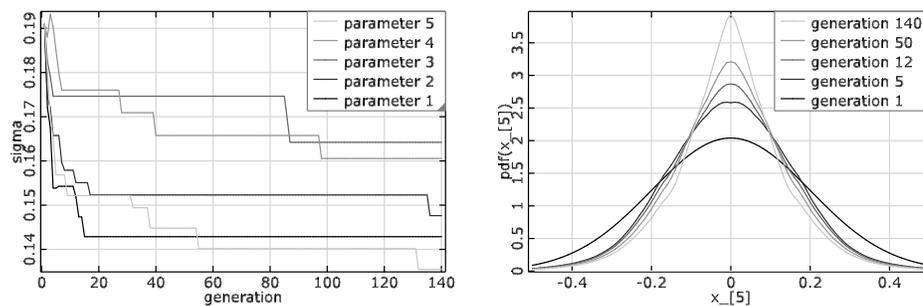


Abbildung 12: TC2: Course of standard deviations of mutation steps (left) and modulated pdf (right) while optimising Rastrigin’s test function.

the Best Fitness Frequency considers all individual of a population for deriving modified strategy parameters, which again are applied to the whole population. The method is recommended for use within GAs, since it is not biased by crossover effects.

Modulating probability density functions introduces adaption on component level, where a random variable for mutation steps is assigned to each optimization variable. Because the strategy evaluates single mutations of individuals, it can only be applied to Evolution Strategies, where no crossover operator can blur the effects of successful mutations.

Literatur

BEYER, H.-G. ; DEB, K.: On Self-Adaptive Features in Real-Parameter Evolutionary Algorithms. In: *IEEE Trans. Evolutionary Computation* 5 (2001), Nr. 3, S. 250–270

BÄCK, T.: *Evolutionary Algorithms in Theory and Practice*. New York : Oxford University Press, 1996

GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1989

- HANSEN, N. ; OSTERMEIER, A.: Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In: *Proceedings of the 1996 IEEE Int. Conf. on Evolutionary Computation*, 1996, S. 312–317
- HANSEN, N. ; OSTERMEIER, A.: Completely Derandomized Self-Adaptation in Evolution Strategies. In: *Evolutionary Computation* 9 (2001), Nr. 2, S. 159–195
- HINTERDING, R. ; MICHALEWICZ, Z. ; EIBEN, A. E.: Adaptation in Evolutionary Computation: a Survey. In: *Proceedings of the Fourth IEEE Conference on Evolutionary Computation, Indianapolis, IN*, 1997, S. 65–69
- HOLLAND, J. H.: *Adaptation in natural and artificial systems*. University of Michigan Press, 1975
- KENNEY, J. F. ; KEEPING, E. S.: Frequency Distributions. In: *Mathematics of Statistics, Pt. 1, 3rd ed.* Princeton, NJ : Van Nostrand, 1962, S. 250–270
- MICHALEWICZ, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. New York : Springer-Verlag, 1992
- MÜLLER, S. D.: *Bio-Inspired Optimization Algorithms for Engineering Applications*. PhD Thesis. Zurich : Swiss Federal Institute of Technology, 2002
- RECHENBERG, I.: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart : Frommann-Holzboog, 1973
- SPEARS, W. M.: *Evolutionary Algorithms: The Role of Mutation and Recombination*. Berlin : Springer-Verlag, 2000
- WRIGHT, S.: The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution. In: *Proceedings of the Sixth Congress on Genetics* Bd. 1, 1992, S. 365